

MANIPULER DES FICHIERS TEXTE

Où l'on saura comment sortir de VI

UTILISER VI / VIM - UN PEU D'HISTOIRE

Wikipedia : vi est un éditeur de texte plein écran écrit par Bill Joy en 1976 sur une des premières versions de la distribution Unix BSD.

Il est présent d'office sur la majorité des systèmes Unix actuels, souvent sous la forme d'un clone du logiciel vi originel.

Que faut-il retenir de ça ?

- VI est présent sur tous les systèmes linux, des fois comme le seul éditeur
- VI a été créé à une époque où ni la souris ni les flèches du clavier n'existaient (OK BOOMER!)
- L'ergonomie des logiciels a quelque peu changé depuis (euphémisme)
- Il est indispensable de connaître le minimum vital pour survivre
- VI est hautement customisable et peut devenir un éditeur très puissant (demande une certaine dose de masochisme)

OUVRIR ET FERMER UN FICHIER

- `vim <file>` : ouvre un fichier

Une fois dans un fichier, il existe deux modes dans VI(m) :

- Le mode commande. Dans ce mode, on peut se déplacer ou taper des commandes. Pour revenir au mode commande, il faut appuyer sur echap.
- Le mode insertion. Dans ce mode, on peut insérer du texte. Nous étudierons quelques basiques du mode insertion après.

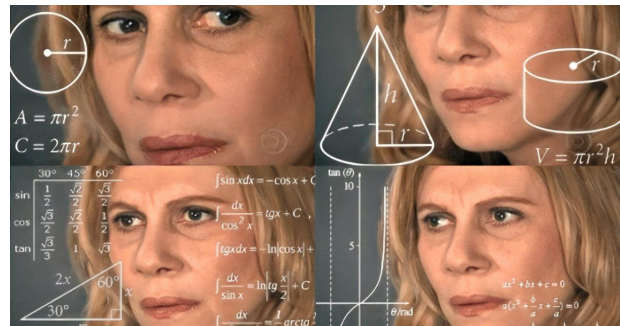
Fermer un fichier :

- En mode commande, faire
 - `:q + enter` : quit
 - `:q! + enter` : force quit (without saving)
 - `:w + enter` : save file
 - `:wq` : save file and quit
 - `:wq!` ou `:x + enter` : save file and force quit



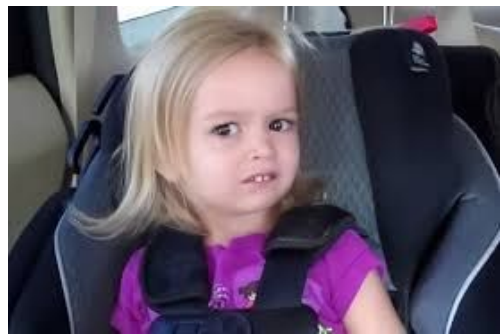
PASSER EN MODE INSERTION

- Une fois positionné à l'endroit qui vous intéresse :
 - i (insert): démarre le mode insertion sur le caractère précédant celui où vous vous trouvez
 - a (append) : démarre le mode insertion après le curseur
 - o : créer une nouvelle ligne en dessous et démarre le mode insertion au début de la ligne
 - r (replace) : remplace un caractère
 - R (REPLACE) : remplace tous les caractères tant que vous continuez à taper et que vous n'appuyez pas sur ECHAP



QUELQUES COMMANDES UTILES

- `u` (undo), annule la précédente action
- `U` (undo / restore), restore previous action
- `dd` : supprime (en vrai ça coupe) la ligne
- `5dd` : supprime la ligne et les 5 suivantes
- `yy` : copie la ligne courante
- `5yy` : copie la ligne et les 5 suivantes
- `p` (paste) : colle le contenu du presse papier



SE DÉPLACER DANS LE FICHER

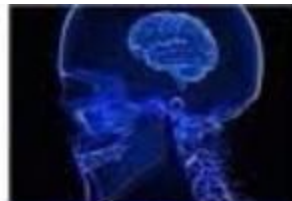


- Si vous utilisez VIM, les flèches directionnelles fonctionnent
- Si jamais vous êtes obligé d'utiliser vi (machine non connectée à internet, très vieille ou devant embarquer le moins de paquets possibles), sachez que les flèches directionnelles risquent de ne pas marcher. Il faudra savoir que :
 - j : down
 - k : up
 - h : left
 - l : right
- G : va à la dernière ligne du fichier
- H : va à la première ligne du fichier
- :5 + enter : va a la 5eme ligne du fichier
- :/toto + enter : cherche toto dans le fichier et positionne le curseur dessus. NB : rappuyer sur enter ira à la prochaine occurrence etc etc..
- \$: positionne le curseur à la fin de la ligne
- 0 : positionne le curseur au début de la ligne

EXERCICE

- Dans votre répertoire utilisateur, créez un fichier `i_love_vim`
- Ouvrez le fichier avec `vim`
- Insérez une nouvelle ligne dans laquelle vous écrirez une petite prose de votre cru
- Sauvegardez et sortez de `VIM`
- Rééditez le fichier
- Insérez une nouvelle phrase avant la précédente
- Annulez la saisie
- Copiez la première phrase
- Collez la 10 fois

google
"how to
exit vim"



force quit
terminal



turn off
computer



slash
computer
with katana



POUR ALLER PLUS LOIN

<https://vim.rtorr.com/>

EXERCICE

- Créez un nouveau fichier texte à partir de <http://metaphorpsum.com/paragraphs/99/99>
- Trouvez la commande permettant d'afficher le contenu d'un fichier depuis le terminal
- Affichez le contenu du fichier depuis le terminal
- Affichez les 20 premières lignes du fichier
- Affichez les 20 dernières lignes du fichier
- Recherchez toutes les occurrences du mot "moment"
- Recherchez toutes les occurrences du mot "moment" et les 2 lignes en dessous

AFFICHER LE CONTENU D'UN FICHER TEXTE

- `cat <file>` #Affiche le contenu d'un fichier dans le terminal
- `head <file>` #Affiche les 10 premières lignes d'un fichier
- `head -n 20 <file>` #Affiche les 20 premières lignes d'un fichier
- `tail <file>` #Affiche les 10 dernières lignes d'un fichier
- `tail -n 20 <file>` #Affiche les 20 dernières lignes d'un fichier
- `tail -f <file>` #Affiche les dernières lignes d'un fichier en temps réel (CTRL+C pour quitter)

RECHERCHER DU TEXTE DANS UN FICHIER / LE FS

- `grep toto <file>` : Cherche la chaîne de caractères toto dans le fichier <file>
- `grep -A3 <file>` : Cherche la chaîne de caractères toto dans le fichier <file> et affiche les 3 lignes suivantes (After)
- `grep -B3 <file>` : Cherche la chaîne de caractères toto dans le fichier <file> et affiche les 3 lignes précédentes (Before)
- `grep -C3 <file>` : Cherche la chaîne de caractères toto dans le fichier <file> et affiche les 3 lignes précédentes et les 3 lignes suivantes
- `grep -v “#” <file>` : Recherche toutes les lignes qui ne contiennent pas #
- `grep -r toto /home` : Cherche la chaîne de caractères toto dans /home et tous ses sous répertoires
- `grep -C3 -r toto /home` : Il est tout à fait possible de combiner plusieurs options